

Shallow Discourse Parsing Using Distributed Argument Representations and Bayesian Optimization

Akanksha

Georgia Institute of Technology
akanksha271@gmail.com

Jacob Eisenstein

Georgia Institute of Technology
jacobe@gmail.com

Abstract

This paper describes the Georgia Tech team’s approach to the CoNLL-2016 supplementary evaluation on discourse relation sense classification. We use long short-term memories (LSTM) to induce distributed representations of each argument, and then combine these representations with surface features in a neural network. The architecture of the neural network is determined by Bayesian hyperparameter search.

1 Introduction

Our approach to discourse relation classification is to combine strong surface features with a distributed representation of each discourse unit. This follows prior work demonstrating that distributed representations can improve generalization for this task (Ji and Eisenstein, 2014; Ji and Eisenstein, 2015; Braud and Denis, 2015). We combine these two disparate representations in a neural network architecture. Our approach is shaped by two main design decisions: the use of long short-term memory recurrent networks (Hochreiter and Schmidhuber, 1997) to induce representations of each discourse unit, and the use of Bayesian optimization (Snoek et al., 2012) for tuning the neural network architecture.

2 System Overview

The overall architecture is shown in Figure 1. The same architecture is used for both explicit and non-explicit relations, but with different parameters. The output of the classifier is a softmax layer, which takes as input a series of dense layers. These dense layers allow nonlinear interactions between surface features and elements of the distributed representations. Dropout is employed

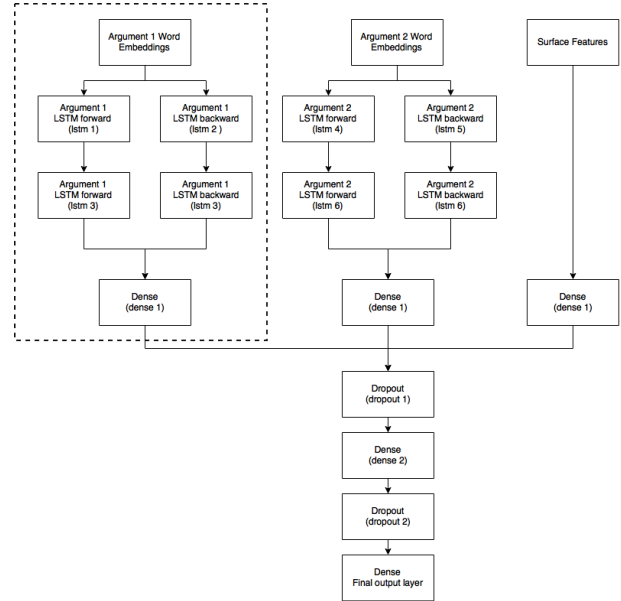


Figure 1: System Architecture

to reduce overfitting (Srivastava et al., 2014). The overall architecture is trained to minimize cross-entropy. The implementation is in Keras (Chollet, 2015), and training takes several hours on a standard CPU. We now describe of the subcomponents of the classifier in detail.

2.1 Distributed representations for discourse units

First, we induce representations for each unit in each discourse relation. This component of the model is shown in the dotted part of Figure 1 for the first discourse argument. Prior work has explored a variety of ways for inducing representations of discourse units, including average pooling (Ji and Eisenstein, 2014; Braud and Denis, 2015) and recursive neural networks on syntactic parse trees (Li et al., 2014; Ji and Eisenstein, 2015). We take a recurrent neural network approach, characterizing

each discourse unit by a recurrently-updated state vector (Li et al., 2015), with the input consisting of pre-trained word embeddings GoogleNews-vectors-negative300.bin from the word2vec page.¹

Specifically, our recurrent architecture is a long short-term memory (LSTM), which uses a combination of gates to better handle long-term dependencies, as compared with the more straightforward recurrent neural network (Hochreiter and Schmidhuber, 1997). Following Graves and Schmidhuber (2005), we employ a bidirectional LSTM, in which each training sequence is presented forwards and backwards to two separate recurrent nets, both of which are connected to the same output layer. We combine the output of these bidirectional LSTMs in a multilayer perceptron with the extracted surface features.

2.2 Surface features

In addition to the distributed representations of the discourse units, we use some of the most successful surface features from prior work. These features are implemented using the Natural Language Toolkit (Bird et al., 2009) and scikit-learn (Pedregosa et al., 2011). In general, these features were inspired by the system from Wang and Lan (2015), which obtained best performance on the PDTB test set in the 2015 shared task (Xue et al., 2015).

2.2.1 Features for explicit relations

Connective Text The connective itself is a strong feature for sense classification of explicit discourse relations (Pitler et al., 2008). This feature alone yields F1 of 0.8862 for our classifier.

Sentiment Value The Vader Sentiment analysis package (Hutto and Gilbert, 2014) was used to calculate sentiment score for both arguments. The feature then reports whether the two arguments have the same sentiment.

Trigrams We used trigram features for the final three words of arg1, and for the first three words of arg2.

2.2.2 Features for non-explicit relations

We used the same **trigrams** features from the explicit relation classifier, as well as the following

Hyperparameter	Range	Best
<i>Number of hidden nodes</i>		
lstm1	64-320	259
lstm2	64-100	75
lstm3	64-320	263
lstm4	64-320	127
lstm5	64-100	89
lstm6	64-320	150
dense1	64-320	269
dense2	64-100	69
<i>Percentage of dropout</i>		
dropout1	0-0.9	0.11
dropout2	0-0.9	0.57
<i>Learning Rate</i>		
SGD	0.001-0.5	0.1549

Table 1: Hyperparameters selected by Spearmint from the provided ranges, for non-explicit discourse relations

additional features on pairs of linguistic elements in arg1 and arg2.

Word Pairs We formed word pairs from the cross product of all words appearing in arg1 and arg2, following much of the prior work in discourse parsing (Marcu and Echihiabi, 2003; Pitler et al., 2009). We then replaced the words in each pair with a cluster identity (Rutherford and Xue, 2014). Specifically, we used the GoogleNews-vectors-negative skipgram word embeddings to form 1000 clusters.

Part-of-Speech Pairs Similarly, we formed part-of-speech pairs from the tags appearing in the two arguments (Rutherford and Xue, 2014).

Production Rules Pairs Using the syntactic analysis of each argument, we form pairs of production rules appearing in the two arguments (Lin et al., 2009).

Adverb Pairs Adverbs are particularly relevant for non-explicit discourse relations, so we compute features from pairs of adverbs appearing the two arguments.

2.3 Hyperparameter tuning

The best set of hyperparameters for the classifiers were found using spearmint (Snoek et al.,

¹<https://code.google.com/archive/p/word2vec/>

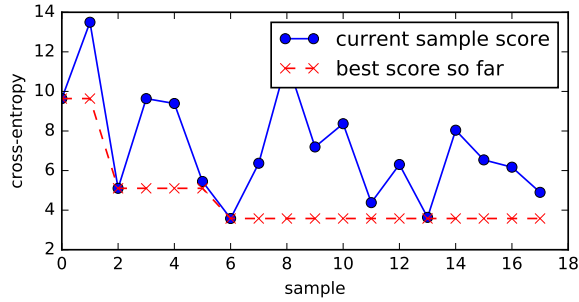


Figure 2: Progress of Bayesian optimization over hyperparameter space

	Feature Type	F1
Non-Explicit	Distributed	0.3485
	+ Argument 2 first 3	0.3872
	+ Argument 1 last 3	0.3044
	+ Word Pairs	0.3672
	+ Parts of Speech	0.3672
	+ Adverbs	0.3979
	+ Inquirer	0.3979
	+ Production Rule	0.4072
Explicit	Distributed	0.3839
	+ Connective	0.8862
	+ Sentiment	0.9029
	+ Argument 2 first 3	0.8816
	+ Argument 1 last 3	0.8983

Table 2: Evaluation as the features are added incrementally to the purely distributed model.

2012), using the `GPEIOptChooser` Markov Chain Monte Carlo search algorithm. This algorithm samples from the space of hyperparameters, while trying to learn a function from hyperparameters to overall performance. We use the cross-entropy (and not the F1) as the overall performance objective; due to the time cost for training the model, we could run only twenty samples, which took several days. The progress of this search is shown in Figure 2, which indicates that the best hyperparameter configuration was identified on the sixth sample. More samples may have yielded better performance, but this was not possible under the time constraints of the shared task. The best set of hyperparameters for non-explicit discourse relation classification are listed in Table 1.

3 Evaluation

Evaluation was performed using the evaluation script provided by the conll16 task organizers. In Table 3, the performance of our system is compared to the best-performing systems from this year’s shared task. Our system was particularly competitive on the blind test set. (The best performance on non-explicit relations on the blind test set was from `ttr`, but this system did not attempt to classify explicit relations, and so did not obtain an overall score for all relations.) This suggests that our approach suffered less from overfitting to the dev data. On the other hand, our system’s performance on explicit relations was further behind the best systems, suggesting the need for additional features to handle this case.

Results are broken down by individual relation types in Table 4, again using the shared task evaluation script. In addition, the contribution of each feature is indicated in Table 2, in which features are incrementally added to a baseline model containing only the distributed representations of each argument.

Acknowledgments We thank the organizers of the shared task for formalizing this evaluation, and we thank Yangfeng Ji for helpful discussions in the initial phase of the project.

System	Dev			Test			Blind		
	All	Explicit	Non-expl.	All	Explicit	Non-expl.	All	Explicit	Non-expl.
tbmihaylov	0.641	0.912	0.403	0.633	0.898	0.392	0.546	0.782	0.345
ecnuc	0.680	0.926	0.464	0.643	0.901	0.409	0.541	0.774	0.342
gtnlp (this paper)	0.639	0.903	0.407	0.609	0.895	0.350	0.543	0.750	0.368

Table 3: Discourse sense classification results, measured by F1, in comparison with the most competitive systems from the shared task.

	Explicit			Non-Explicit		
	<i>precision</i>	<i>recall</i>	<i>F1</i>	<i>precision</i>	<i>recall</i>	<i>F1</i>
Micro-Average	0.9029	0.9029	0.9029	0.4072	0.4072	0.4072
Comparison.Concession	1.0000	0.0833	0.1538	1.0000	0.0000	0.0000
Comparison.Contrast	0.9387	0.9563	0.9474	0.2391	0.2619	0.2500
Contingency.Cause.Reason	0.8235	0.6829	0.7467	0.3714	0.1688	0.2321
Contingency.Cause.Result	1.0000	0.8421	0.9143	0.3714	0.1688	0.2321
Contingency.Condition	0.9778	0.9362	0.9565	-	-	-
EntRel	-	-	-	0.5143	0.7535	0.6113
Expansion.Alternative	0.8571	1.0000	0.9231	-	-	-
Expansion. Alternative.Chosen alternative	1.0000	0.8333	0.9091	1.0000	0.0000	0.0000
Expansion.Conjunction	0.9286	0.9891	0.9579	0.3298	0.5122	0.4013
Expansion.Instantiation	1.0000	1.0000	1.0000	0.4783	0.2292	0.3099
Expansion.Restatement	0.3750	0.2621	0.3086	0.3750	0.2621	0.3086
Temporal.Asynchronous.Precedence	0.9608	1.0000	0.9800	0.2857	0.0800	0.1250
Temporal.Asynchronous.Succession	1.0000	0.6667	0.8000	1.0000	0.0000	0.0000
Temporal.Synchrony	0.6842	0.9420	0.7927	1.0000	0.0000	0.0000

Table 4: Dev set evaluation for explicit and non-explicit (Implicit, EntRel, AltLex) discourse relations

References

- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*. O’Reilly Media, California.
- Chloé Braud and Pascal Denis. 2015. Comparing word representations for implicit discourse relation classification. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 2201–2211, Lisbon, September.
- François Chollet. 2015. Keras. <https://github.com/fchollet/keras>.
- Alex Graves and Jürgen Schmidhuber. 2005. Frame-wise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*.
- Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the Association for Computational Linguistics (ACL)*, Baltimore, MD.
- Yangfeng Ji and Jacob Eisenstein. 2015. One vector is not enough: Entity-augmented distributional semantics for discourse relations. *Transactions of the Association for Computational Linguistics (TACL)*, June.
- Jiwei Li, Rumeng Li, and Eduard Hovy. 2014. Recursive deep models for discourse parsing. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.
- Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. 2015. When are tree structures necessary for deep learning of representations? In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 2304–2314, Lisbon, September.

- Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 343–351, Singapore.
- Daniel Marcu and Abdessamad Echihabi. 2003. An unsupervised approach to recognizing discourse relations. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 368–375.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind Joshi. 2008. Easily identifiable discourse relations. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 87–90, Manchester, UK.
- Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Association for Computational Linguistics (ACL)*, Suntec, Singapore.
- Attapol T Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.
- Jasper Snoek, Hugo Larochelle, and Ryan Prescott Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Neural Information Processing Systems*.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Jianxiang Wang and Man Lan. 2015. A refined end-to-end discourse parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 17–24, Beijing, China, July. Association for Computational Linguistics.
- Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol T Rutherford. 2015. The CoNLL-2015 shared task on shallow discourse parsing. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*.